



Flex Bulgaria
User Group

Using Enhanced Constraints

by Fabian Vercuiel





- **Introduction**
- **Description of Constraints**
- **From Parent Child to Sibling Relative Constraints**
- **Workflow benefits with Enhanced Constraints**
- **Creating typical website layouts with different types of constraints**
- **Integration**
- **Links**



- **Overview of familiar Flex Constraints**
 - **Typical Absolute Constraint Layouts**
 - Demo
 - **Typical Relative Constraint Layouts**
 - Demo
 - **Real world Scenario using combined Constraint Layouts**
 - Demo
- **Introduction to Enhanced Constraints**

Typical Absolute Constraint Layouts

- Have fixed/absolutely positioned coordinates
- Is either positioned with x / y or top / bottom / left / right anchors
- Can constrain to the edges of its Absolute parent container
- Does not resize dynamically
- Child elements are drawn at the 0,0 coordinates of their containers when not positioned, multiple non-positioned items will stack on top of each other
- Elements are not Vertically or Horizontally relative to one another



Flex Bulgaria
User Group

DEMO

Typical Relative Constraint Layouts

- Have percentage/relative positioned coordinates
- Is positioned with padding, center-constraints, align, gap styles etc.
- Resizes dynamically according to the size of its parent if set to a percentage size
- Child elements are drawn relative to each other vertically or horizontally according to the layout properties of their parent containers.
- Elements are restricted to their relative relationships so overlapping positioning of child elements with relative containers is not possible
- Have the possibility of setting even automatic spacing between children



Flex Bulgaria
User Group

DEMO

Real world Scenario using combined typical Constraint Layouts

- Has all the features mentioned above about Absolute and Relative constraints.
- Typical work flow example of this would be using relative constraints for forms, grids, lists and any extensive repeating elements in either a horizontal or vertical direction; then using absolute containers and elements for the general layout with some relative style properties like center constraints.
- Allows for anchoring relative items to a canvas at the expense of adding additional nested containers.
- This method of combining constraint types makes it possible to create advanced parent-child layouts.



Flex Bulgaria
User Group

DEMO



Enhanced Constraints:

- The idea behind this new approach to creating layouts is to expand upon the constraint-based layout system you are all familiar with. Allowing constraining of elements to not be dependent on the nesting of containers.
- Enhanced constraints allow you to constrain child controls to each other, the edges of the parent containers and stretches and reflows as the Canvas container resizes.
- Enhanced constraints adds a 3rd sizing option called “content”, alongside fixed and percentage based sizing. This allows a container to calculate the size of the parent according to the dynamic reflow of its children.



- **ConstraintColumn and ConstraintRow**
 - Description
 - 3 Sizing options
 - Demo
- **Constraint**
 - Enhanced Constraints
 - Demo
 - Center Constraints
 - Baseline Constraints



Description of Constraints

ConstraintColumn and ConstraintRow

Description

- Can only be created inside a Canvas
- There are 2 properties – ConstraintColumn and ConstraintRow.
- Each property will contain child objects which are faceless objects, something like invisible guidelines or ghost subdivisions.
- The origin of these child objects when defining the size properties start from 0,0 of the Canvas.
- ConstraintRows are defined top to bottom and ConstraintColumns are defined left to right in the order that they were defined.
- You can have only have only one constraint region eg. only creating ConstraintColumns



3 Sizing Options

- **Fixed size**
 - The space taken up by that region is a fixed pixel size
- **Percentage size**
 - The space taken up by that region is a percentage of the overall space available.
- **Content size**
 - The space taken up by that region is dictated by the content contained in that space. As the size of the content changes, so does the region.
- **MinWidth and MaxWidth**
 - Constraint columns/rows will be able to set min/max sizes to control how much they grow or shrink. As a region grows below its minimum size, the parent Canvas sprouts scrollbars to show hidden content.



Flex Bulgaria
User Group

DEMO



Enhanced Constraints

- Positioning is handled with left, right, top and bottom.
- Controls can be positioned inside and/or across the constraint regions which were specified with ConstraintRow and ConstraintColumn.
- Positioning properties contain the id of either row or column with the offset value separated by a semicolon.
- A control needs to have paired constraint regions if you wish for the content to reflow.
- The hardest concept to understand initially when creating Enhanced Constraint layouts is that of if you have a column named "col1" with width="200" and you specify on your control the position **left="col1:0"** and **right="col1:0"** you are asking the control to draw at 0px and end at 200px of the region, you don't have to create 2 columns, each column has left, right, top, bottom regions. Don't think of columns as just lines with 1 edge.



Flex Bulgaria
User Group

DEMO



Center Constraints

- Two styles are available -horizontalCenter and verticalCenter
- If a center constraint is set, that constraint is evaluated relative to the constraint region specified, be it a containers edges or a ConstraintRow/Column.

```
<mx:Button horizontalCenter="0" />
<!-- The Button is placed directly in the horizontal center of its container -->
  <mx:Button verticalCenter="0" />
  <!-- The Button is placed directly in the vertical center of its container -->
    <mx:Button horizontalCenter="col1:0"/>
    <!-- The Button is placed in the horizontal center of ConstraintColumn -->
      <mx:Button verticalCenter="20"/>
      <!-- The Button is placed 20 pixels from the direct center of its container -->
```



Baseline Constraints

- Setting a baseline constraint means that the user is specifying an offset from the top edge of the constraint region and the controls baseline.
- By offering a baseline constraint we can now have multiple components constrain to a single shared line.

```
<mx:Button baseline="20" />
<!-- The Button is placed such that its baseline is 20px from the top edge -->
<mx:Button baseline="row1:0" />
<mx:Label baseline="row1:0" />
<mx:LinkButton baseline="row1:0" />
<!-- The 3 controls are all baseline constrained to the same constraint row -->
```



Flex Bulgaria
User Group

From Parent Child to Sibling Relative Constraints

- **What is a Parent Child Constraint Layout**
- **What is a Sibling Relative Constraint Layout**



From Parent Child to Sibling Relative Constraints

What is a Parent Child Constraint

- This layout is used conventionally in Flex.
- Layout structure and relationship between elements done by nesting containers.
- The idea behind this style of development is that elements are nested within each other for logical and visual separation. eg. It makes sense having a text box inside a container you wish to use.
- Child element is always dependant on the positioning properties of its parent
- You'd nest a inner container inside a parent container so that you can use constraint edges of a inner container instead of positioning each and every element from the parent or for utilizing different layout logic. ie. Parent is a Canvas and child is a VBox



Flex Bulgaria
User Group

DEMO



Flex Bulgaria From Parent Child to Sibling Relative Constraints

What is a Sibling Relative Constraint

- The layout style is applied to the Enhanced Constraints workflow
- Sibling Relative means that there is no deep dependency between nested containers and that in most cases elements are level to each other in the parent container. Each element is absolutely positioned with the possibility to anchor to one of the invisible relatively positioned constraint regions.
- The idea is that each sibling element has minimal amount of influence from its parent(s), usually only from the Application and Parent Canvas. All elements are on the same level and positioning is completely flexible with little effort.
- Siblings have the ability to be completely relative to each other, they can all share the same constraint edges whereas in ParentChild you have to sometimes change between various containers to achieve a desired layout.



Flex Bulgaria
User Group

DEMO

Workflow Benefits with Enhanced Constraints

- Designers will be able to comprehend the concept of Enhanced Constraints much more than the (as they see it) nested gibberish. Designers are used to being able to move elements freely as they do in Image Authoring applications. They want to be able to communicate this easily to developers. They can see ConstraintColumns/Rows as guidelines in Photoshop/Flash etc.and explain their designs easier that way.
- If all your elements share the same groups of constraints moving elements will be simpler, moving 1 constraint will cause the rest of the design to flow seamlessly saving you time.
- Enhanced constraints allow you to create advanced relative reflowing content driven layouts, this would only be possible in Parent Child layouts by using Bindable objects and other methods that take time and experience.



- Is currently integrated with Flex Beta 3, replaces grid constraints entirely.
- Design view support for Enhanced Constraints will not ship with Flex 3 since they were too far from a stable version.
- You can expect Enhanced constraints to be supported throughout the future flex releases including support with Thermo.



- Flex Wiki: <http://flexwiki.adobe.com>
“Flex 3 Details - Enhanced Constraints”
- Flex Labs: <http://labs.adobe.com>
“ Flex 3:Feature Introductions: Enhanced Constraints”
- iamdeepa Blog: <http://iamdeepa.com/blog/?p=17>
“Where did the FlexBuilder Advanced Constraints UI go?”



Flex Bulgaria
User Group

Links

QUESTIONS?



Flex Bulgaria
User Group

Links

THANK YOU